IN THE CLAIMS

Please amend the claims as follows:

Claim 1 (Currently Amended): A software/hardware language model conversion method for converting a first code described in a software description language to a second code described in a hardware description language, the method comprising:

converting the first code to the second code;

detecting from the second code a plurality of processes ~~from the second code, the processes corresponding to~~ that correspond to a plurality of parallel procedures in the first code~~, which assign values to predetermined shared variable; and~~;

identifying which ones of the plurality of processes assign values to an identical shared variable; and

generating a value solving process for the detected processes corresponding to the parallel procedures, wherein the value solving process includes pairs of a data signal and an assignment timing signal from each of the detected processes as an input, and includes any one of data signals corresponding to a change of the assignment timing signal, as an output.

Claim 2 (Original): The method according to claim 1, wherein said second code includes a component hierarchical structure, wherein said value solving process includes a first value solving process which is generated at higher level in the component hierarchical structure and a second value solving process which is generated at lower level than that of the first value solving process, and further includes:

connecting the generated first and second value

solving processes each other.

Claim 3 (Original): The method according to claim 2, wherein the second value solving process outputs an assignment timing signal which is changed when a value of the data signal is updated to said first value solving process connected to the second value solving process, in addition to said data signal.

Claim 4 (Original): The method according to claim 2, further comprising:

connecting an input to the second value solving process to an input to the first value solving process; and

removing the second value solving process.

Claim 5 (Currently Amended): A software/hardware language model conversion method for converting a first code described in a software description language to a second code described in a hardware description language, the method comprising:

detecting from the first code a plurality of processes that correspond to a plurality of parallel procedures in the second code ~~that assign values to a predetermined shared variable from the first code~~;

identifying which ones of the plurality of parallel procedures assigns values to an identical shared variable;

converting the first code to the second code; and

generating a value solving process for the detected processes corresponding to the parallel procedures, wherein the value solving process includes pairs of a data signal and an assignment timing signal from each of the detected processes as an input, and includes any one of data signals corresponding to a change of the assignment timing signal, as an output.

Claim 6 (Original): The method according to claim 5, wherein said second code includes a component hierarchical structure, wherein said value solving process includes a first value solving process which is generated at higher level in the component hierarchical structure and a second value solving process which is generated at lower level than that of the first value solving process, and further includes:

connecting the generated first and second value solving processes each other.

Claim 7 (Original): The method according to claim 6, wherein the second value solving process outputs an assignment timing signal which is changed when a value of the data signal is updated to said first value solving process connected to the second value solving process, in addition to said data signal.

Claim 8 (Original): The method according to claim 6, further comprising:

connecting an input to the second value solving process to an input to the first value solving process; and

removing the second value solving process.

Claim 9 (Currently Amended): A software/hardware language model conversion method for converting a first code described in a software description language to a second code described in a hardware description language, the method comprising:

converting the first code to the second code;

detecting from the second code a plurality of processes ~~from the second code, the processes corresponding to~~ that correspond to a plurality of parallel procedures in the first code ~~to call a predetermined parallel procedure~~;

identifying which ones of the plurality of processes assign values to an identical shared variable; and

generating a procedure call solving process which exclusively controls a call operation to the parallel procedure by the detected process, wherein the procedure call solving process includes pairs of ~~an~~ a call timing signal and an argument data signal, from each of the detected processes, as an input, and includes terminal signal and a return value signal, as an output to the processes, from the called procedure.

Claim 10 (Original): The method according to claim 9, wherein said second code includes a component hierarchical structure, wherein said procedure call solving process includes a first procedure call solving process which is generated at higher level in the component hierarchical structure and a second procedure call solving process which is generated at lower level than that of the first procedure call solving process, and further includes:

connecting the generated first and second procedure call solving processes each other.

Claim 11 (Original): The method according to claim 10, further comprising:

connecting an input to the second procedure call solving process to an input to the first procedure call solving process; and

removing the second procedure call solving process.

Claim 12 (Currently Amended): A software/hardware language model conversion method for converting a first code described in a software description language to a second code described in a hardware description language, the method comprising:

detecting from the first code a plurality of procedures ~~from the first code, which call a predetermined parallel procedure~~ that correspond to a plurality of parallel procedures in the second code;

identifying which ones of the plurality of parallel procedures assign values to an identical shared variable;

converting the first code to the second code; and

generating a procedure call solving process which exclusively controls an operation of a plurality of processes corresponding to the detected procedures, wherein the procedure call solving process includes pairs of a call timing signal and an argument data signal from each of the detected processes, as an input, and includes a terminal signal and a return value signal, as an output to the processes, from the called procedure.

Claim 13 (Original): The method according to claim 12, wherein said second code includes a component hierarchical structure, wherein said procedure call solving process includes a first procedure call solving process which is generated at higher level in the component hierarchical structure and a second procedure call solving process which is generated at lower level than that of the first procedure call solving process, and further includes:

connecting the generated first and second procedure call solving processes each other.

Claim 14 (Original): The method according to claim 13, further comprising:

connecting an input to the second procedure call solving process to an input to the first procedure call solving process; and

removing the second procedure call solving process.

Claim 15 (Currently Amended): A software/hardware language model conversion method for converting a first code described in a software description language to a second code described in a hardware description language, the method comprising:

converting the first code to the second code;

detecting from the second code a plurality of processes ~~from the second code, the processes corresponding to~~ that correspond to a plurality of parallel procedures in the first code~~, which assign values to a predetermined shared variable~~;

identifying which ones of the plurality of processes assign values to an identical shared variable;

generating a value solving process for the detected processes corresponding to the parallel procedures, wherein the value solving process includes pairs of a data signal and an assignment timing signal from each of the detected processes, as an input, and includes any one of data signals corresponding to a change of the assignment timing signal, as an output;

converting the value solving process relating to a procedure call between parallel programs to a procedure call solving process wherein the procedure call solving process includes pairs of an call timing signal and an argument data signal from each of the detected processes, as an input, and includes a terminal signal and a return value signal, as an output to the processes, from the called procedure.

Claim 16 (Currently Amended): A software/hardware language model conversion method for converting a first code described in a software description language to a second code described in a hardware description language, the method comprising:

detecting from the first code a plurality of processes that correspond to a plurality of parallel procedures ~~that assign values to a predetermined shared variable from the first~~ in the second code;

identifying which ones of the plurality of processes assign values to an identical shared variable; and

converting the first code to the second code;

generating a value solving process for the detected processes corresponding to the parallel procedures, wherein the value solving process includes pairs of a data signal and an assignment timing signal from each of the detected processes, as an input, and includes any one of data signals corresponding to a change of the assignment timing signal, as an output, the data signal indicating a value of the shared variable; and

converting the one of the value solving process relating to a procedure call between parallel programs to a procedure call solving process, wherein the procedure call solving process includes pairs of a call timing signal and an argument data signal from each of the detected processes, as an input, and includes a terminal signal and a return value signal, as an output to the processes, from the called procedure.

Claim 17 (Currently Amended): A computer program product comprising:

a computer storage medium and a computer program code mechanism embedded in the computer storage medium for causing a computer to convert a first code described in a software description language to a second code described in a hardware description language, the computer code mechanism comprising:

a computer code device configured to convert the first code to the second code;

a computer code device configured to detect from the second code a plurality of processes ~~from the second code, the processes corresponding to~~ that correspond to a plurality of parallel procedures in the first code~~, which assign values to a predetermined shared variable~~;

a computer code device configured to identity which ones of the plurality of processes assign values to an identical shared variable; and

a computer code device configured to generate a value solving process for the detected processes corresponding to the parallel procedures, wherein the value solving process includes pairs of a data signal and an assignment timing signal from each of the detected processes as an input, and includes any one of data signals corresponding to a change of the assignment timing signal, as an output.

Claim 18 (Original): The computer program product according to claim 17, wherein said second code includes a component hierarchical structure, wherein said value solving process includes a first value solving process which is generated at higher level in the component hierarchical structure and a second value solving process which is generated at lower level than that of the first value solving process, and further includes:

a computer code device configured to connect the generated first and second value solving processes each other.

Claim 19 (Original): The computer program product according to claim 18, wherein the second value solving process outputs an assignment timing signal which is changed when a value of the data signal is updated to said first value solving process connected to the second value solving process, in addition to said data signal.

Claim 20 (Original): The computer program product according to claim 18, further comprising:

a computer code device configured to connect an input to the second value solving process to an input to the first value solving process; and

a computer code device configured to remove the second value solving process.

Claim 21 (Currently Amended): A computer program product comprising:

a computer storage medium and a computer program code mechanism embedded in the computer storage medium for causing a computer to convert a first code described in a software description language to a second code described in a hardware description language, the computer code mechanism comprising:

a computer code device configured to detect from the first code a plurality of processes that correspond to a plurality of parallel procedures ~~that assign values to a predetermined shared variable from the first code~~ in the second code;

a computer code device configured to identify which ones of the plurality of processes assign values to an identical shared variable;

a computer code device configured to convert the first code to the second code; and

a computer code device configured to generate a value solving process for the detected processes corresponding to the parallel procedures, wherein the value solving process includes pairs of a data signal and an assignment timing signal from each of the detected processes as an input, and includes any one of data signals corresponding to a change of the assignment timing signal, as an output.

Claim 22 (Original): The computer program product according to claim 21, wherein said second code includes a component hierarchical structure, wherein said value solving process includes a first value solving process which is generated at higher level in the component hierarchical structure and a second value solving process which is generated at lower level than that of the first value solving process, and further includes:

a computer code device configured to connect the generated first and second value solving processes each other.

Claim 23 (Original): The computer program product according to claim 22, wherein the second value solving process outputs an assignment timing signal which is changed when a value of the data signal is updated to said first value solving process connected to the second value solving process, in addition to said data signal.

Claim 24 (Original): The computer program product according to claim 22, further comprising:

a computer code device configured to connect an input to the second value solving process to an input to the first value solving process; and

a computer code device configured to remove the second value solving process.

Claim 25 (Currently Amended): A computer program product comprising:

a computer storage medium and a computer program code mechanism embedded in the computer storage medium for causing a computer to convert a first code described in a software description language to a second code described in a hardware description language, the computer code mechanism comprising:

a computer code device configured to convert the first code to the second code

a computer code device configured to detect from the second code a plurality of processes ~~from the second code, the processes corresponding to~~ that correspond to a plurality of procedures ~~to call a predetermined parallel procedure~~ in the first code;

a computer code device configured to identify which ones of the plurality of processes assign values to an identical shared variable; and

a computer code device configured to generate a procedure call solving process which exclusive controls a call operation to the parallel procedure by the detected process, wherein the procedure call solving process includes pairs of ~~an~~ a call timing signal and an argument data signal, from each of the detected processes, as an input, and includes terminal signal and a return value signal, as an output to the processes, from the called procedure.

Claim 26 (Original): The computer program product according to claim 25, wherein said second code includes a component hierarchical structure, wherein said procedure call solving process includes a first procedure call solving process which is generated at higher level in the component hierarchical structure and a second procedure call solving process which is generated at lower level than that of the first procedure call solving process, and further includes:

a computer code device configured to connect the generated first and second procedure call solving processes each other.

Claim 27 (Original): The computer program product according to claim 26, further comprising:

a computer code device configured to connect an input to the second procedure call solving process to an input to the first procedure call solving process; and

a computer code device configured to remove the second procedure call solving process.

Claim 28 (Currently Amended): A computer program product comprising:

a computer storage medium and a computer program code mechanism embedded in the computer storage medium for causing a computer to convert a first code described in a software description language to a second code described in a hardware description language, the computer code mechanism comprising:

a computer code device configured to detect from the first code a plurality of procedures ~~from the first code, which call~~ that correspond to a plurality of ~~predetermined~~ parallel procedures in the second code;

a computer code device configured to identify which ones of the plurality of procedures assign values to an identical shared variable;

a computer code device configured to convert the first code to the second code; and

a computer code device configured to generate a procedure call solving process which exclusive controls an operation of a plurality of processes corresponding to the detected procedures, wherein the procedure call solving process includes pairs of a call timing signal and an argument data signal from each of the detected processes, as an input, and includes a terminal signal and a return value signal, as an output to the processes, from the called procedure.

Claim 29 (Original): The computer program product according to claim 28, wherein said second code includes a component hierarchical structure, wherein said procedure call solving process includes a first procedure call solving process which is generated at higher level in the component hierarchical structure and a second procedure call solving process which is generated at lower level than that of the first procedure call solving process, and further includes:

a computer code device configured to connect the generated first and second procedure call solving processes each other.

Claim 30 (Original): The computer program product according to claim 29, further comprising:

a computer code device configured to connect an input to the second procedure call solving process to an input to the first procedure call solving process; and

a computer code device configured to remove the second procedure call solving process.

Claim 31 (Currently Amended): A computer program product comprising:

a computer storage medium and a computer program code mechanism embedded in the computer storage medium for causing a computer to convert a first code described in a software description language to a second code described in a hardware description language, the computer code mechanism comprising:

a computer code device configured to convert the first code to the second code;

a computer code device configured to detect <u>from the second code</u> a plurality of processes ~~from the second code, the processes corresponding to~~ <u>that correspond to</u> a plurality

of parallel procedures in the first code~~, which assign values to a predetermined shared variable~~;

a computer code device configured to identify which ones of the plurality of processes assign values to an identical shared variable;

a computer code device configured to generate a value solving process for the detected processes corresponding to the parallel procedures, wherein the value solving process includes pairs of a data signal and an assignment timing signal from each of the detected processes, as an input, and includes any one of data signals corresponding to a change of the assignment timing signal, as an output;

a computer code device configured to convert the value solving process relating t a procedure call between parallel programs to a procedure call solving process wherein the procedure call solving process includes pairs of an call timing signal and an argument data signal from each of the detected processes, as an input, and includes terminal signal and a return value signal, as an output to the processes, from the called procedure.

Claim 32 (Currently Amended): A computer program product comprising:

a computer storage medium and a computer program code mechanism embedded in the computer storage medium for causing a computer to convert a first code described in a software descript on language to a second code described in a hardwire description language, the computer code mechanism comprising:

a computer code device configured to detect from the first code a plurality of processes that correspond to a plurality of parallel procedures ~~that assign values to a predetermined shared variable from~~ in the ~~first~~ second code;

A1

a computer code device configured to identify which ones of the plurality of processes assign values to an identical shared variable;

a computer code device configured to convert the first code to the second code;

a computer code device configured to generate a value solving process for the detected processes corresponding to the parallel procedures, wherein the value solving process includes pairs of a data signal and an assignment timing signal from each of the detected processes, as an input, and includes any one of data signals corresponding to a change of the assignment timing signal, as an output, the data signal indicating a value of the shared variable; and

a computer code device configured to convert the one of the value solving process relating to a procedure call between parallel programs to a procedure call solving process, wherein the procedure call solving process includes pairs o a call timing signal and an argument data signal from each of the detected processes, as an input, and includes a terminal signal and a return value signal, as an output to the processes, from the called procedure.